

**Amendments to the Claims**

Please amend Claims 1, 8, and 15. The Claim Listing below will replace all prior versions of the claims in the application:

**Claim Listing**

1. (Currently amended) A computer apparatus for managing and sharing engineering data for chemical engineering processes and plants, comprising:
  - an editor defining (i) class views and (ii) a composite class view of the defined class views, given one or more software applications of interest and each given software application having a respective data model or data view, for each said given software application, the editor providing a class view of the respective data model;
  - the editor consolidating said class views to form a composite class views, the consolidating of said class views resulting in the creation of said composite class view being an amalgamation and a rationalization of the individual class views, and the class views being retained in a given software application domain terminology; and
  - a data server executed by a digital processor and instantiating a multi-tier data model, there being a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the core conceptual data model, the class views effectively being one tier of the multi-tier data model, the composite class view effectively being a second tier of the multi-tier data model and the core conceptual data model effectively being a third tier, the multi-tier data model having links between corresponding attributes across tiers, the multi-tier data model providing management and sharing of engineering data of the given software applications with other process and plant engineering applications, and enhancing process engineering and plant operations.
2. (Previously presented) The computer apparatus as claimed in Claim 1 further comprising an amalgamator that synthesizes the class views, the composite class view and the core conceptual data model into a consolidated multi-tier data model.

3. (Previously presented) The computer apparatus as claimed in Claim 1 further comprising a mapper that links the core conceptual data model attributes to the composite class view and the composite class view attributes to class views, and provides a one-to-one mapping between an attribute in the composite class view and a route in the core conceptual data model to corresponding given software applications from which the attribute in the composite class view originated.
4. (Previously presented) The computer apparatus as claimed in Claim 3 wherein each class view is represented in terms from the respective given software application, and said given software application is able to access data from the core conceptual data model.
5. (Previously presented) The computer apparatus as claimed in Claim 1 wherein the class views, the composite class view and the core conceptual data model are represented by object oriented programming elements.
6. (Previously presented) The computer apparatus as claimed in Claim 5 wherein certain object oriented programming elements are defined by classes; and wherein the editor enables user creation and editing of definitions of classes.
7. (Previously presented) The computer apparatus as claimed in Claim 6 wherein the editor employs an Extensible Markup Language.
8. (Currently amended) A method of data modeling, comprising the computer implemented steps of:
  - (a) forming a multi-tier data model with links between corresponding attributes across tiers, a first tier being formed by:
    - for each of multiple given software applications of interest and having a respective data model, providing a practitioner's view of the given software application using a respective class view of the respective data model;

a second tier being formed by consolidating class views into a composite class view, the consolidation of said class views resulting in the creation of said composite class view being an amalgamation and a rationalization of the individual class views, and the class views being retained in a given software application domain terminology; and

a third tier being formed by forming a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the core conceptual data model; and

(b) sharing, via the multi-tier data model, engineering data of the given software applications with other process and plant engineering routines.

9. (Previously presented) The method as claimed in Claim 8 wherein the second tier is formed by synthesizing the class views into the composite class view.
10. (Previously presented) The method as claimed in Claim 8 wherein the step of forming a multi-tier data model further comprises producing a one-to-one mapping between an attribute in each class view to the composite class view, and a one-to-one mapping between an attribute in the composite class view and a route in the core conceptual data model to corresponding given software applications from which the attribute in the composite class view originated.
11. (Previously presented) The method as claimed in Claim 8 wherein the step of providing a practitioner's view includes, in each class view, representing the respective data model in terms from the respective given software application.
12. (Previously presented) The method as claimed in Claim 8 further comprising the step of representing at least one of the class views, the composite class view and the core conceptual data model in terms of object oriented programming elements.

13. (Previously presented) The method as claimed in Claim 12 wherein certain object oriented programming elements are defined by classes; and  
enabling user creation and edition of definitions of classes.
14. (Previously presented) The method as claimed in Claim 13 wherein the step of enabling user creation and edition includes employing Extensible Markup Language interfaces.
15. (Currently amended) A computer program product comprising:
  - (a) a computer readable medium that manages engineering data; and
  - (b) a set of computer program instructions encoded on the computer readable medium, the set of computer program instructions when executed on a computer causing the computer to:
    - provide a respective class view for each of plural given software applications of interest and having a respective data model, each class view being of the respective data model;
    - form a composite class view from the class views, the consolidation of said class views resulting in the creation of said composite class view being an amalgamation and a rationalization of the individual class views, and the class views being retained in a given software application domain terminology;
    - form a conceptual model;
    - form a consolidated multi-tier data model from the class views, the composite class view and the conceptual model; and
    - via, the consolidated multi-tier data model, provide sharing of engineering data of the given software applications with other process and plant engineering applications.
16. (Previously presented) The computer program product of Claim 15, wherein the consolidated multi-tier data model insulates the given software applications from changes in the conceptual model.

17. (Previously presented) The computer program product of Claim 15, wherein the consolidated multi-tier data model is insulated from changes in the given software applications.
18. (Previously presented) The computer program product of Claim 15, wherein the consolidated multi-tier data model provides an application independent and normalized data model where the composite class view is application independent.
19. (Previously presented) The computer program product of Claim 15, wherein the consolidated multi-tier data model comprises an editor and a class store, the class store providing an interface to the respective class views, the composite class view, and the conceptual model to share data between the consolidated multi-tier data model and the given software applications.
20. (Previously presented) The computer program product of Claim 19, wherein the editor and the class store use an Extensible Markup Language, and wherein the composite class view provides for the class views remaining in the given software application domain terminology.